

McCaw Cellular Takes the NextStep

McCaw managers use Next's Enterprise Objects Framework to rapidly produce new applications that help the company remain competitive in an enterprise that handles billing for millions of calls per day

By John G. Maguire

M*cCaw Cellular Communications Inc. has become the largest cellular telephone company in the United States by responding quickly to changing market conditions and customer needs. It's no secret that competition in the cellular phone business is fierce and highly dependent on the computers that run every aspect of the business. Although AT&T Corp. purchased McCaw last year for \$11 billion, McCaw's information technology managers are still focused on remaining nimble enough to retain their market leadership.*

Casting about for solutions to help the company develop applications on a more timely basis, McCaw's manager of applied architecture, Jonathan Weeks, and his staff in Kirkland, Wash., began an evaluation in late 1994 of Enterprise Objects Framework (EOF) from Next Computer Inc.

"Think about what happened with MCI," says Weeks. "MCI announced its 'Friends & Family' calling plan. It was actually a discount based on a special billing method. AT&T tried to react but couldn't revise its billing software fast enough to offer a similar product. The result? MCI took market share from AT&T."

The cellular telephone business is still a young industry—barely a decade old. About three years ago, its growth curve exploded. Now roughly 17,000 new cellular phone customers are added each day. McCaw's share is 3,400 new customers per day. McCaw managers realize that the company needs to rapidly produce new applications that can handle the implementation of new policies, billing plans, and special first-time discounts. And these applications must be available across an enormous enterprise that handles and bills millions of calls per day in 105 markets.

The problem for McCaw's IT department is not only maintaining crucial operational code such as billing software, but being agile enough to add new features to it in a timely fashion. But there's a formidable obstacle. According to Weeks, the size of McCaw's source code is huge and hard to modify. "Execution size for our application code is up around 8 to 10 megabytes," he says. "Adding new features and functions when you have 100,000 lines of code can be very difficult."

Objective Advantage

EOF is both a product and an architecture from Next that allows object-oriented applications written in the Objective C language to access and use relational database management systems (RDBMS). It's a type of product some analysts call an "object-relational enabler." It runs under the NextStep operating system on Intel microprocessor-based PCs, as well as on workstations from Hewlett-Packard Co. (and soon will be available on Sun Microsystems Inc. systems). It can also run on non-NextStep platforms, including SunOS, Sunsoft Inc. Solaris, HP-UX, and Digital Equipment Corp.'s OSF/1, if these processors are running Next's Portable Distributed Objects



Enterprise Objects Framework

Next Computer Inc.
900 Chesapeake Dr.
Redwood City, CA 94065
800-879-6398, 415-366-0900
fax: 415-780-3714
e-mail: nextanswers@next.com
(Type "help" in the subject line)
Web site: <http://www.next.com>

Price: Enterprise Objects Framework sells for \$299. This object-oriented software toolkit is an adjunct to the NextStep operating system, which sells for \$795.

Systems: EOF Version 1.1 runs on Motorola, Intel, HP-PA RISC, and SPARC processors. It comes bundled with Sybase System 10 and Oracle 7 adapters.

(PDO). PDO is a server-based product that sells for roughly \$5,000.

"Enterprise Objects Framework allows you to keep your existing investment in RDBMS technology and all the skill sets associated with it," says Hugh Bishop, manager of emerging technology research at the market research firm Aberdeen Group Inc. in Boston. "But you get to use the advantages of an object-oriented development environment."

According to Bishop, those advantages include rapid development, code reuse, easier maintenance, and better modeling of the real world. "Fundamentally, what object-relational enablers do," says Bishop, "is automate the process of storing complex objects in a relational database."

Relational databases are fundamental to the cellular phone business. Everything McCaw does for its customers is done with, or with the aid of, database software. Customers are signed up, their credit is verified, and their service is activated with a point-of-sale application. And they are billed by computer.

Whenever McCaw's customers are

driving around in their cars, their contact with their cellular service providers is also computer mediated. If a call is interrupted, or if the caller wants to reach the cellular phone company for anything, he or she dials *611 and is connected with a cellular phone customer-service representative. A NextStep software application linked to the caller's records guides the representative through every step of the call.

"We have coverage maps, calling areas, a list of area codes a customer can dial for free, instructions on how to unlock the phone, pictures of most common phones, and a graphical picture of the exact bill the customer has in front of him [or her]," explains Weeks.

The Missing Link?

EOF allows developers to build reusable business and application-level objects that are independent of underlying databases. According to Bishop, EOF and similar products are the "missing link" for deploying object-oriented applications on a wide-scale distributed basis.

EOF consists of three modules: a modeler, a run-time module, and a set of adapters. The modeler helps the developer build "business objects" and create a mapping structure that sits between the business objects and the databases used by the application. The run-time module allows these objects to run on the NextStep environment. Two database adapters—for Oracle and Sybase—comprise the third module. These adapters provide an interface between the mapping structure and an individual database.

EOF essentially transforms the traditional two-level relational database model into a three-level model. The standard database consists of rows of data more or less "hardwired" directly to the fields in a screen for an individual application. EOF, however, inserts an intermediate level between the data and the user screen. This "business-logic" level contains processing logic for each application, as well as rules and procedures shared across an enterprise. It also contains what Next calls the "enterprise objects," which embody the company's rules and procedures.

These mid-level objects can be created, recombined, and reused independent of either user interface or database-access objects. When EOF is combined with Next's PDOs, it can allow production of client-server applications that serve the entire enterprise, and reduce software costs.

Custom adapters can be written so that EOF can access data from relational databases other than Sybase and Oracle, such as SofDesign's QuickBase. "I've even seen people who have written flat-file adapters or network feed adapters," says Tyler Gingrich, vice president of engineering at Vanguard Software Corp. in Cincinnati. "It doesn't matter. Once the data bubbles up and goes through this entity-relationship modeling process, it can be manipulated like a regular NextStep object."

EOF can also connect simultaneously to multiple databases, by the same or different vendors, according to Next. For large companies this can be key.

Evaluation and Usage

McCaw's evaluation of EOF began in late 1994. McCaw officials were severely skeptical when they began their rigorous testing of EOF. "Much of what Next has done we think a great deal of," says Ingvar Petursson, McCaw's vice president and chief information officer, "but not everything. We were not happy that Next's earlier releases came without a persistence framework or portable distributed objects, and that we had to write our own in-house."

Until recently, only ad hoc tools were available to marry object code to RDBMS. Similar products, according to the Aberdeen Group, include Easel Corp.'s Synchronicity, which manages Smalltalk objects; HP's Oadapter, which manages C++ and Smalltalk objects; ParcPlace Systems Inc.'s Visual Works 2.0, which provides Smalltalk applications with RDBMS access; Persistence Software's tools for C++ applications; and UniSQL's product UniSQL/M, a multidatabase manager that handles applications written in C, C++, and Smalltalk.

For nearly three years, McCaw had

used a combination of Next's DB Kit and an in-house tool it called "persistence framework." In order to test



**The stability of
EOF and the increase
in production it
spurred impressed
McCaw's managers**

EOF, McCaw developers benchmarked the product against its own set of in-house tools. At the start of the test, Weeks set his stopwatch and ordered his staff to replicate a number of McCaw's applications under EOF. "We were very apprehensive about the fact that it was a new product," Weeks recalls (EOF was introduced in late June 1994). "But it has been very stable."

Nearly three months later, the results were "very impressive," he says, "a major gain in productivity." Weeks estimates that the new object development process is a five-fold improvement. McCaw's CIO Petursson places the gain at seven-fold.

"Under EOF, our lines of application code shrank by 80 percent, and developer productivity went way up," Weeks says. The productivity gain required the use of all of Next's tools, including EOF, as well as the application-building tools Project Builder and Interface Builder.

A further plus, Weeks reports, is the stability of the product. "The first alpha release we had of EOF was more stable than DB Kit was in its third and fourth full releases. Basically, Next had rethought the whole problem, completely scrapped the architecture of DB Kit, and started over."

The question for the company then

became how to completely and rapidly switch to the new software system. It was clear that new code would be written on EOF—but what about existing applications? Would it be worth the cost and effort of redoing them? This was a nontrivial question. Streamlining the company's ability to write and maintain account-handling applications is crucial, because McCaw intends to differentiate itself by its first-class customer service.

Although cellular phone service is growing rapidly, there are problems. While the customer base currently increases by 30 to 40 percent a year, the dropout rates are also high. The industry loses 2 percent of its customers a month on average, according to Tod Wolfenbarger, a McCaw spokesman. "We pay a lot of money to have these people come on," he says. "How do we keep the customers we have happy?"

New Projects

The billing operation for McCaw is currently outsourced to a Cincinnati Bell Information Systems Inc. (CBIS) division in Maitland, Fla. First on the list of projects to be tackled using the new framework is bringing most of the corporate billing structure in-house. According to Weeks, "CBIS's system works on an IBM mainframe. The company is dependable, but it isn't reacting as fast or as cheaply as our people want it to." A further worry, Weeks says, is that CBIS provides billing services to many of McCaw's competitors. Until McCaw brings its billing in-house, any innovations in billing will be risky. "If we want some new twist in our billing plan, we'd rather have it in-house and not immediately lose the advantage to other clients of the company we're outsourcing to."

At the end of 1994, McCaw answered the "million-dollar question" of how much to port to EOF. Everything would port to EOF during the next year.

"This was not an easy decision. We've got two-and-a-half years' experience with our in-house program," says Petursson. "On the other hand, we rewrote a key application that was 70,000 lines of C code, and under EOF it shrank to under 10,000 lines."

As the product continues to evolve, even its boosters admit the existence of a few irritations. One small "technical nit," says Gingrich, is what he believes is a slightly awkward requirement: an "object controller" needs to be used to update objects. Gingrich contends that there is a performance hit when using EOF because the data retrieved must be massaged before it can show up in the form of an object. The performance hit, though, is "not significantly bigger" than those found in competing solutions, he adds.

EOF allows rapid object-oriented programming in conjunction with a relational database, and the use of those objects across an enterprise. Judging by McCaw's developers' intensely positive reactions to it, EOF seems likely to open up more commercial accounts for Next. ■

John G. Maguire, formerly director of Boston University's science journalism program, writes about computers, biotechnology, and science in Boston.

**"Under EOF, our lines
of application code
shrank by 80 percent,
and developer
productivity went
way up"**

Reprinted with permission from Open Computing, April 1995.
Copyright 1995 by Open Computing.
All rights reserved. For reorders, call 612-582-3800.



For further information about NeXT Computer, Inc. products and services, please call 1-800-TRY-NeXT.